

Applying the Meaning-Text Theory Model to Text Synthesis with Low- and Middle Density Languages in Mind

Leo WANNER^{a,b} and François LAREAU^b

^a*Institució catalana de recerca i estudis avançats (ICREA)*

^b*Universitat Pompeu Fabra, Barcelona*

The linguistic model as defined in the Meaning Text Theory (MTT) is, first of all, a language production model. This makes MTT especially suitable for language engineering tasks related to synthesis: text generation, summarization, paraphrasing, speech generation, and the like. In this article, we focus on text generation. Large scale text generation requires substantial resources, namely grammars and lexica. While these resources are increasingly compiled for high density languages, for low- and middle density languages often no generation resources are available. The question on how to obtain them in a most efficient way becomes thus imminent. In this article, we address this question for MTT-oriented text generation resources.

1. Introduction

With the progress of the state of the art in Computational Linguistics and Natural Language Processing, language engineering has become an increasingly popular and pressing task. This is in particular true for applications in which text synthesis constitutes an important part – among other things, automatic text generation, summarization, paraphrasing, and machine translation.

Large coverage text synthesis in general requires substantial resources for each of the languages involved. While for high density languages, more and more resources are available, many of the low- and middle density languages are still not covered. This may be due to the lack of reference corpora, the lack of specialists knowledgeable in the field and in the language in question, or other circumstances. The question which must obviously be answered when a text synthesis application is to be realized for a low- or middle density language for which no text synthesis resources are available as yet: How can these resources be obtained in the most rapid and efficient way? This obviously depends on the exact kind of resources needed and thus, to a major extent, on the application and the linguistic framework underlying the implementation of the given system that addresses the application.

In this article, we focus on one case of text synthesis: the natural language text generation. We discuss the types of resources required for a text generator based on the *Meaning Text Theory*, MTT [1, 2, 3, and 4]. MTT is one of the most common rule-based linguistic theories used for text generation. This is not by chance: MTT's model is synthesis (rather than analysis, or parsing) oriented and it is formal enough to allow

for an intuitively clear and straightforward development of resources needed for generation.

The remainder of the article is structured as follows. In Section 2, we present the basics of the MTT-model and the kind of resources it requires. Section 3 contains a short overview of the tasks implied in text generation. In Section 4, text generation from the perspective of MTT is discussed. Section 5 elaborates on the principles for efficient compiling of generation resources and Section 6 discusses how resources for new languages can be acquired starting from already existing resources. Section 7 addresses the important problem of the evaluation of grammatical and lexical resources for MTT-based generation. Section 8, finally, contains some concluding remarks.

As is to be expected from an article summarizing an introductory course on the use of Meaning-Text Theory in text generation, most of the material is not novel. The main information sources used for Section 2 have been [1, 2, 3, and 5]. For Sections 4, 5, and 7, we draw upon [6, 7, 8, and 9] and, in particular, on [10], which is reproduced in parts in the abovementioned sections. A number of other sources of which we also make use are mentioned in the course of the article.

2. Meaning-Text Theory and its Linguistic Model

MTT interprets language as a rule-based system which defines a many-to-many correspondence between an infinite countable multitude of meanings (or semantic representations, SemRs) and an infinite multitude of texts (or phonetic representations, PhonRs); cf., e.g., [1, 2, and 3]:

$$\bigcup_{i=1}^{\infty} \text{SemR}_i \leftrightarrow \bigcup_{j=1}^{\infty} \text{PhonR}_j$$

This correspondence can be described and verified by a formal model – the Meaning-Text Model (MTM). In contrast to many other linguistic theories such as *Systemic Functional Linguistics* [11, 12], *Functional Linguistics* [13]; *Cognitive Linguistics* [14, 15], *Role and Reference Grammar* [16], etc., MTT is thus in its nature a formal theory.

An MTM is characterized by the following five cornerstone features:

- (i) it is *stratificational* in that it covers all levels of a linguistic representation: semantic, syntactic, morphological and phonetic, with each of the levels being treated as a separate stratum;
- (ii) it is *holistic* in that it covers at each stratum all structures of the linguistic representation: **at the semantic stratum**, the *semantic* (or *propositional*) *structure* (*SemS*) which encodes the content of the representation in question, the *communicative* (or *information*) *structure* (*CommS*) which marks the propositional structure in terms of salience, degree of acquaintance, etc. to the author and the addressee, and the *rhetorical structure* (*RhetS*) which defines the style and rhetorical characteristics that the author wants to give the utterance under verbalization; **at the syntactic stratum**, the syntactic structure (*SyntS*), the *CommS* which marks the syntactic structure, the *co-referential structure* (*CorefS*) which contains the co-reference links between entities of the syntactic structure denoting the same object, and the *prosodic structure* (*ProsS*) which specifies the intonation contours, pauses, emphatic stresses, etc.; **at the morphological stratum**, the *morphological structure* (*MorphS*) which encodes the word order and internal morphemic organization of word

forms, and the ProsS; **at the phonetic stratum**, the *phonemic structure* (PhonS) and the ProsS.¹

- (iii) it is *dependency-oriented* in that the fundamental structures at each stratum are dependency structures or are defined over dependency structures;
- (iv) it is *equivalence-driven* in that all operations defined within the model are based on equivalence between representations of either the same stratum or adjacent strata;
- (v) it is *lexicalist* in that the operations in the model are predetermined by the features on the semantic and lexical units of the language in question – these features being stored in the lexicon.

Depending on the concrete application in which we are interested, more or less strata, more or less structures are involved. As mentioned in Section 1, in this article, we focus on automatic text generation, i.e., on written, rather than on spoken texts. Therefore, the more detailed definition of the notion of the Meaning-Text Model can discard the phonetic representation, such that the definition reads as follows:

Definition: *MTT Model, MTM*

Let **SemR** be the set of all well-formed meaning (or semantic) representations, **SyntR** the set of all well-formed syntactic representations, **MorphR** the set of all well-formed morphological representations and **Text** the set of all texts of the language $\mathcal{Q}$ such that any $\text{SemR} \in \mathbf{SemR}$, any $\text{SyntR} \in \mathbf{SyntR}$, and any $\text{MorphR} \in \mathbf{MorphR}$ is defined by the corresponding basic structure and a number of auxiliary structures: $\text{SemR} = \{\text{SemS}, \text{CommS}, \text{RhetS}\}$, $\text{SyntR} = \{\text{SyntS}, \text{CommS}, \text{CorefS}, \text{ProsS}\}$, and $\text{MorphR} = \{\text{MorphS}, \text{ProsS}\}$. Let the basic structures be directed labeled graphs of different degree of freedom such that any directed relation r between a node a and node b in a given structure, $a \xrightarrow{r} b$, expresses the dependency of b on a of type r . Let furthermore a and b be semantic units in a $\text{SemS} \in \text{SemR}$, lexical units in a $\text{SyntS} \in \text{SyntR}$, and morphemes in a $\text{MorphS} \in \text{MorphR}$. Then, the MTM of $\mathcal{Q}$ over $\mathbf{SemR} \cup \mathbf{SyntR} \cup \mathbf{MorphR} \cup \mathbf{Text}$ is a quadruple of the following kind

$$\text{MTM} = (\text{M}_{\text{SemSynt}}, \text{M}_{\text{SyntMorph}}, \text{M}_{\text{MorphText}}, \mathcal{D}), \text{ such that}$$

A grammar module M_i (with $i \in \{\text{SemSynt}, \text{SyntMorph}, \text{MorphText}\}$) is a collection of equivalence rules, $\mathcal{D}$ is the set of dictionaries (lexica) of $\mathcal{Q}$ and the following conditions hold:

$$\begin{aligned} \forall \text{SemR}_i \in \mathbf{SemR}: (\exists \text{SyntR}_j \in \mathbf{SyntR}: \text{M}_{\text{SemSynt}}(\text{SemR}_i, \mathcal{D}) &= \text{SyntR}_j) \\ \forall \text{SyntR}_i \in \mathbf{SyntR}: (\exists \text{MorphR}_j \in \mathbf{MorphR}: \text{M}_{\text{SyntMorph}}(\text{SyntR}_i, \mathcal{D}) &= \text{MorphR}_j) \\ \forall \text{MorphR}_i \in \mathbf{MorphR}: (\exists \text{Text}_j \in \mathbf{Text}: \text{M}_{\text{MorphText}}(\text{MorphR}_i, \mathcal{D}) &= \text{Text}_j) \end{aligned}$$

The syntactic and morphological strata are further split into a “deep”, i.e., content-oriented, and a “surface”, i.e., “syntax oriented” substratum, such that in total we have to deal with six strata; Figure 1 shows the resulting picture.

¹ In what follows, we will call the semantic, syntactic, morphological, and phonemic structures the “basic structures” of the corresponding stratum.

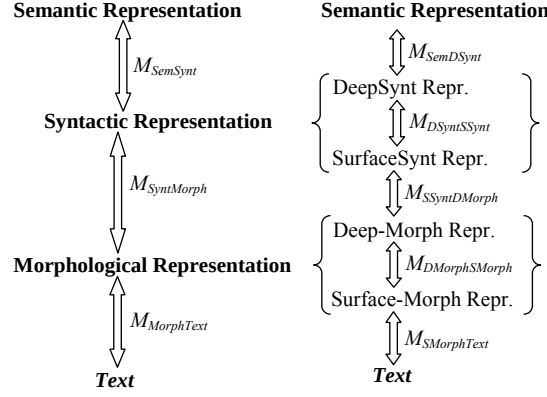


Figure 1: Meaning-Text Linguistic Model

In the remainder of the section, we briefly describe the individual strata, the modules of the MTM and the dictionaries the modules make use of.

2.1. Definition of the Strata in an MTM

As already outlined above, a linguistic representation at a given stratum of the MTM is defined by its basic structure and the co-referential, communicative, rhetorical, and prosodic structures as complementary structures that are defined over the corresponding basic structure. The rhetorical structure can also be treated as part of the *context of situation* [17] and the prosodic structure is irrelevant for written texts; we leave them thus aside in our rough introduction to MTT and focus on the first three structures, which are essential for our application.

2.1.1. Basic structures at the individual strata of an MTM

Let us introduce, in what follows, the definition of the basic structures of an MTM that play a role in text generation: the semantic, the deep-syntactic, the surface-syntactic, and the deep-morphological structures. Surface-morphological structures are similar to deep-morphological structures except that they already have all morphological contractions, elisions, epenthesis and morph amalgamation performed. Therefore, we do not discuss them here.

Definition: *Semantic Structure (SemS)*

Let S_{Sem} and R_{Sem} be two disjoint alphabets of a given language $\mathcal{L}$, where S_{Sem} is the set of *semantemes* of $\mathcal{L}$ and R_{Sem} is the set of semantic relation names $\{1, 2, \dots\}$. A semantic structure, Str_S , is a quadruple (G, α, β, D_S) over $S_{Sem} \cup R_{Sem}$ with

- $G = (N, A)$ as a directed acyclic graph, with the set of nodes N and the set of directed arcs A ;
- α : as the function that assigns to each $n \in N$ an $s \in S_{Sem}$
- β : as the function that assign to each $a \in A$ an $r \in R_{Sem}$
- D_S as the semantic dictionary with the semantic valency of all $s \in S_{Sem}$

such that for any $\alpha(n_i) - \beta(a_k) \rightarrow \alpha(n_j) \in Str_S$ the following restrictions hold:

1. $\beta(a_k)$ is in the semantic valency pattern of $\alpha(n_i)$ in D_S
2. $\exists n_m, a_l: \alpha(n_i) - \beta(a_l) \rightarrow \alpha(n_m) \wedge \beta(a_k) = \beta(a_l) \Rightarrow a_k = a_l \wedge n_j = n_m$

The conditions ensure that a SemS is a predicate-argument structure.

Although SemS is language-specific, it is generic enough to be isomorphic for many utterances in similar languages. Consider, e.g., the following eight sentences:²

1. Eng. *Orwell has no doubts with respect to the positive effect that his political engagement has on the quality of his works.*
2. Ger. *Orwell hat keine Zweifel, was den positiven Effekt seines politischen Engagements auf seine Arbeiten angeht*
lit. 'Orwell has no doubts what the positive effect of his political engagement on his works concerns'.
3. Rus. *Orvell ne somnevaetsja v tom, čto ego političeskaja angažirovannost' položitel'no vlijaet na kačestvo ego proizvedenij*
lit. 'Orwell does not doubt in that that his political engagement positively influences [the] quality of his works'.
4. Serb. *Orvel ne sumnja u to da njegov politički angažman deluje povoljno na kvalitet njegovih dela*
lit. 'Orwell does not doubt in that that his political engagement acts positively on [the] quality of his works'.
5. Fr. *Orwell n'a pas de doute quant à l'effet positif de son engagement politique sur la qualité de ses œuvres*
lit. 'Orwell does not have doubt with respect to the positive effect of his political engagement on the quality of his works'.
6. Sp. *Orwell no duda que sus actividades políticas tienen un efecto positivo en la calidad de sus obras.*
lit. 'Orwell does not doubt that his political activities have a positive effect on the quality of his works'.
7. Cat(alan). *Orwell no dubta que les seves activitats polítiques tenen un efecte positiu en la qualitat de les seves obres*
lit. 'Orwell does not doubt that the his political activities have a positive effect on the quality of the his works'.
8. Gal(ician). *Orwell non dubida que as súas actividades políticas teñen un efecto positivo na calidade das súas obras*
lit. 'Orwell does not doubt that the his political activities have a positive effect on quality of-the his works'.

Some of the sentences differ with respect to their syntactic structure significantly, yet their semantic structures are isomorphic, i.e., they differ merely with respect to node labels. Figure 2 shows the English sample. The number 'i' of a semanteme stands for the 'i-th sense' of the semanteme's name captured by this semanteme.

Note that the structure in Figure 2 is simplified in that it does not contain, for instance, temporal information which corresponds to a specific verbal tense at the syntactic strata; it does not decompose the comparative semanteme 'better.5'; etc.

To obtain the other seven semantic structures, we simply need to replace the English semantemes by the semantemes of the corresponding language. This is not to say that the semantic structures of equivalent sentences are always isomorphic. They

² The original French sentence is from [3].

can well differ both within a single language (cf. [18] for semantic paraphrasing within one language) and between languages – for instance, when the distribution of the information across the same semantic representation is different as in the case of the Indo-European vs. Korean/Japanese politeness system [7].

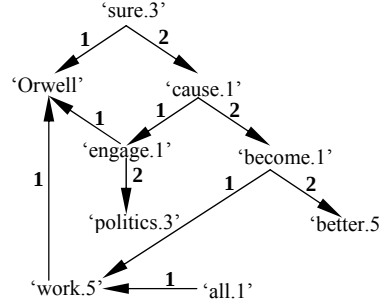


Figure 2: Semantic structure of sentence 1

Definition: *Deep-syntactic Structure* (DSyntS):

Let L_D , R_{DSynt} and G_{sem} be three disjunct alphabets of a given language $\mathcal{Q}$ where L_D is the set of deep lexical units (LUs) of $\mathcal{Q}$, R_{DSynt} is the set of DSynt relations $\{I, II, III, \dots\}$ of $\mathcal{Q}$ and G_{sem} the set of semantic grammemes of $\mathcal{Q}$.

A DSyntS, Str_{DSynt} , is a quintuple $(G, \alpha, \beta, \gamma, D_L)$ over $L_D \cup R_{DSynt} \cup G_{sem}$, with

- $G = (N, A)$ as a dependency tree, with the set of nodes N and the set of arcs A
- α as the function that assigns to each $n \in N$ an $l \in L_D$
- β : as the function that assigns to each $a \in A$ an $r_{ds} \in R_{DSynt}$
- γ : as the function that assigns to each $\alpha(n)$ a set of semantic grammemes
- D_L : as the dictionary with the syntactic valency of all $l \in L_D$

such that for any $\alpha(n_i) - \beta(a_k) \rightarrow \alpha(n_j) \in Str_{DSynt}$ the following restrictions hold:

1. $\beta(a_k)$ is in the syntactic valency pattern of $\alpha(n_i)$ in D_S
2. $\exists n_m, a_l: \alpha(n_i) - \beta(a_l) \rightarrow \alpha(n_m) \wedge \beta(a_k) = \beta(a_l) \Rightarrow a_k = a_l \wedge n_j = n_m$

The set of deep-lexical LUs contains the LUs of the vocabulary of the language $\mathcal{Q}$ to which two types of artificial LUs are added and three types of LUs are excluded. The added LUs include: (i) symbols denoting *lexical functions* (LFs), (ii) *fictitious lexemes*.

LFs are a formal means to encode lexico-semantic derivation and restricted lexical co-occurrence (i.e., collocations); cf., among others, [19, 20, 21, and 22]: SMOKE $\Rightarrow$ SMOKER, SMOKER $\Rightarrow$ HEAVY $[\sim]$, SMOKE $\Rightarrow$ HAVE $[a \sim_N]$, ...³ Each LF carries a functional label such as S_1 , Magn and Oper₁: $S_1(\text{SMOKE}) = \text{SMOKER}$, Magn(SMOKER) = HEAVY, Oper₁(SMOKE_N) = HAVE.

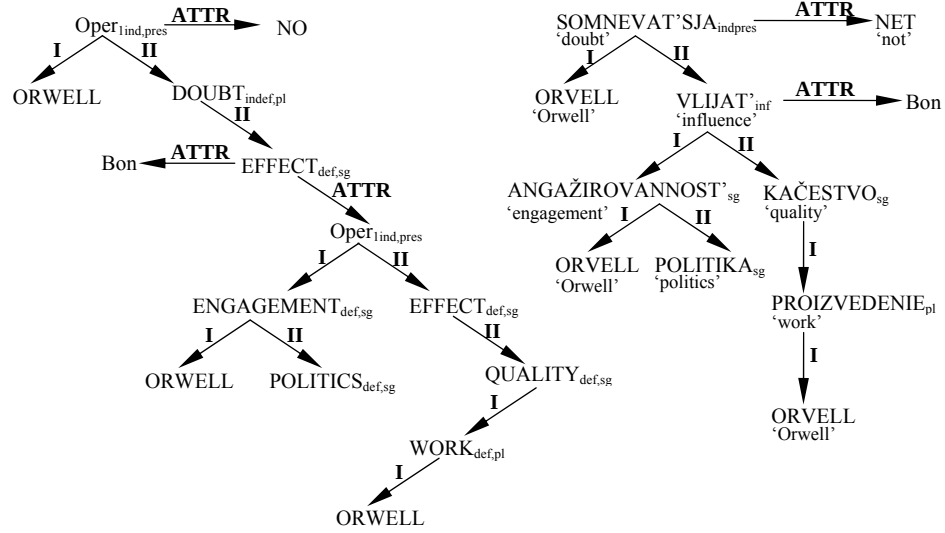
Fictitious lexemes represent idiosyncratic syntactic constructions in $\mathcal{Q}$ with a predefined meaning – as, for instance, $\langle X_{\text{count.noun}} N_{\text{number}} \rangle$ meaning ‘roughly N of X’ in Russian.

³ ‘ $\sim$ ’ stands for the LU in question.

The excluded LUs involve: (i) structural words (i.e., auxiliaries, articles, and governed prepositions), (ii) substitute pronouns, i.e., 3rd person pronouns, and (iii) values of LFs.

Semantic grammemes are obligatory and regular grammatical significations of inflectional categories of LUs; for instance, *nominal number*: singular, dual, ...; *voice*: active, passive, ...; *tense*: past, present, future; *mood*: indicative, imperative, ...; and so on.

Compared to SemS, DSyntS is considerably more language-specific, although it is abstract enough to level out surface-oriented syntactic idiosyncrasies; cf. Figure 3 for the DSyntS of the sentences 1 (English) and 3 (Russian) from above. Oper₁ and Bon are names of LFs. The subscripts are the grammemes that apply to the lexeme in question.



Divergences between semantically equivalent DSyntSs can be of lexical, syntactic, or morphological nature [23, 6, and 7].

Definition: *Surface-syntactic Structure (SSyntS)*

Let L , R_{SSynt} and G_{sem} be three disjunct alphabets of a given language $\mathcal{L}$, where L is the set of lexical units (LUs) of $\mathcal{L}$, R_{SSynt} is the set of SSynt relations and G_{sem} the set of semantic grammemes.

A SSyntS, Str_{SSynt} , is a quintuple $(G, \alpha, \beta, \gamma, D_L)$ over $L \cup R_{SSynt} \cup G_{sem}$, with

- $G = (N, A)$ as a dependency tree, with the set of nodes N and the set of arcs A
- α as the function that assigns to each $n \in N$ an $l \in L$
- β : as the function that assigns to each $a \in A$ an $r_{ss} \in R_{SSynt}$
- γ : as the function that assigns to each $\alpha(n)$ a set of semantic grammemes
- D_L : as the dictionary with the syntactic valency of all $l \in L$

such that for any $\alpha(n_i) - \beta(a_k) \rightarrow \alpha(n_j) \in Str_{SSynt}$ the following restrictions hold:

1. $\beta(a_k)$ is in the syntactic valency pattern of $\alpha(n_i)$ in D_L
2. $\exists n_m, a_l: \alpha(n_i) - \beta(a_l) \rightarrow \alpha(n_m) \wedge \beta(a_k) = \beta(a_l) \Rightarrow a_k = a_l \wedge n_j = n_m$

Consider in Figure 4 as example of an SSyntS, the surface-syntactic structure of sentence 7 (i.e., Catalan) from above.

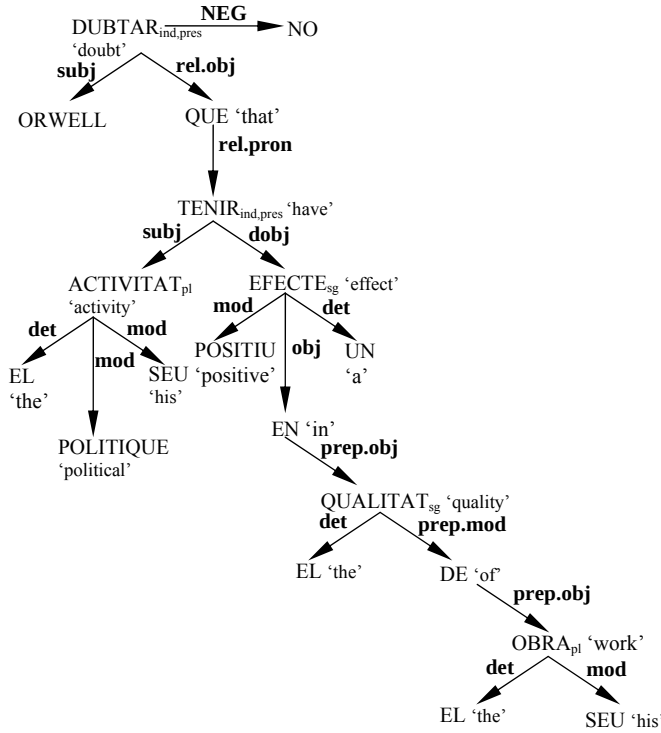


Figure 4: SSyntS of the sample sentence 7 above

Definition: *Deep-morphological structure* (DMorphS)

Let L and G_{sem} be disjoint alphabets of a given language $\mathcal{L}$, where L is the set of lexical units (LUs) of $\mathcal{L}$ and G_{sem} the set of semantic grammemes.

A DMorphS, Str_{DMorph} , is a quadruple $(G, \alpha, \kappa, \gamma, \rho)$ over $L \cup G_{sem}$, with

- $G = (N, <)$ as a chain, with the set of nodes N and the precedence relation ' $<$ ',
- α as the function that assigns to each $n \in N$ an $l \in L$
- κ as the function that defines over N the set of constituents C
- γ : as the function that assigns to each $\alpha(n)$ a set of semantic grammemes
- ρ : as the function that defines for each pair $c_i, c_j \in C$ a precedence order relation.

The main function of a DMorphS is thus to provide a linearization of the dependency tree as given in the corresponding SSyntS. Following [24], we assume that in contrast to the preceding basic structures, DMorphS also reflects a phrase structure.⁴ Consider, e.g., the Catalan DMorphS for our sample sentence:

ORWELL_{sg} < [[NO < DUBTAR_{3p.sg.ind.pres}] < [QUE < [EL_{fem.pl} < SEU_{fem.pl} < ACTIVITAT_{fem.pl}
 < POLITIQUE_{fem.pl}] < TENIR_{3p.pl.ind.pres} < [UN_{masc.sg} < EFFECTE_{masc.sg} < POSITIU_{masc.sg} < EN
 < EL_{fem.sg} < QUALITAT_{fem.sg} < [DE < [EL_{fem.pl} < SEU_{fem.pl} < OBRA_{fem.pl}]]]]]

Figure 5: DMorphS of the sentence 7 above

2.1.2. Co-referential structure

A co-referential structure (CorefS) consists of a set of (bidirectional) co-references between different nodes of a given basic structure that denote the same object. CorefS is defined over a DSyntS or an SSyntS; given that a SemS is an acyclic graph (rather than a tree), there is no need to repeat nodes that denote the same object: all relations can refer to one single node without that well-formedness conditions of the propositional structure are violated. As a consequence, CorefS is defined between deep lexeme nodes (in a DSyntS) and between surface lexeme nodes (in a SSyntS). For instance, in Figure 3, CorefS_{DSynt} would contain ‘N_i[ORWELL] ↔ N_j[ORWELL]’ and ‘N_i[EFFECT] ↔ N_j[EFFECT]’ (with i and j being the number of the corresponding nodes in the propositional structure).

2.1.3. Communicative Structure

The Communicative Structure (CommS) (or *information structure*) is central to text generation in that it determines the distribution of the information across a sentence and ensures the cohesion between the sentences. In contrast to commonly known proposals concerning information structure [26, 27, 28, and 29], CommS in an MTM is a very rich multidimensional structure that is defined over a basic structure:

Definition: *Communicative structure* (CommS)

Let Str_i be a basic structure at the stratum S_i . The communicative structure $CommS_i$ defined over Str_i is a set of possibly overlapping labeled areas, such that the labels belong to eight different dimensions D_i ($i = 1, 2, \dots, 8$) as specified below and the following restriction holds: If in the $CommS_i$ two areas overlap, then, the labels of these areas belong to different dimensions. The set of dimensions is given by the following eight dimensions:

- D1: {Theme, Rheme, Specifier}
- D2: {Given, New}
- D3: {Focalized, Non-focalized}
- D4: {Foregrounded, Backgrounded, Neutral}
- D5: {Presupposed, Non-Presupposed}
- D6: {Emphasized, Non-Emphasized}
- D7: {Unitary, Articulated}
- D8: {Signaled, Performed, Communicated}

⁴ For some languages, a more complex phrase-like structure is needed to determine the word order. Consider, for instance, the *topological field model* for German [25]; for the treatment of the field model in MTT, see [24].

For a detailed introduction to CommS in general and to all eight dimensions, see [5]; to give an impression of how important CommS is, it suffices to focus on the first four dimensions.

2.1.3.1. The Thematicity Dimension

The three-partite *Thematicity*-dimension is the primary dimension in text generation. *Rheme* identifies what is stated; *Theme* identifies about what Rheme is stated; and *Specifier* captures the circumstances under which the statement is made.⁵ In complex structures, the Theme or Rheme elements may be recursive, i.e., contain embedded Thematicity partitions; cf. the Thematicity structure of SemS from Figure 2 in Figure 6, where Rheme is subsequently subdivided into more fine-grained Theme-Rheme partitions:

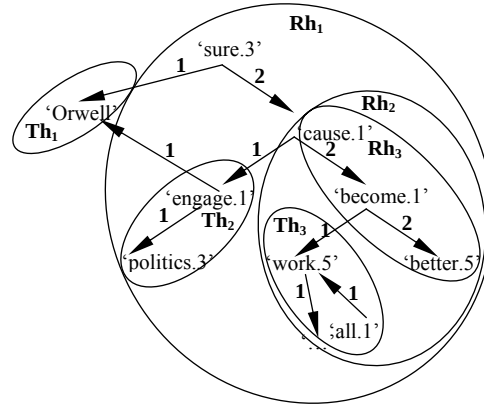


Figure 6: Thematicity distribution in SemR of sentence 1 (the outgoing arrow of ‘work.5’ points to ‘Orwell’)

Thematicity is so crucial for text generation because it provides information for the determination of the basic sentence structure: roughly speaking, the Theme-element is predestined to become the subject of a clause and the Specifier-elements attributive constructions.

2.1.3.2. The Givenness Dimension

The bipartite *Givenness*-dimension with *Given* and *New* as its elements is equally immediately reflected in the sentence structure and the morpho-syntactic features of its elements. *Given* is what the author presents as known to the addressee, and *New* is what the author presents to be unknown to the addressee. Figure 7 illustrates the influence of Givenness on generation. The left side contains simple semantic structures with different distributions of Given [G] and New [N]. The right side shows the effect of each distribution at the surface of the corresponding sentence.

$['\text{friends}']_N \leftarrow 1 - ['\text{present}']_N \rightarrow 2 \rightarrow ['\text{show}']_N$	$\equiv$ Some friends present / make a presentation of a show.
$['\text{friends}']_N \leftarrow 1 - ['\text{present}']_G \rightarrow 2 \rightarrow ['\text{show}']_N$	$\equiv$ Some friends make the presentation of a show.
$['\text{friends}']_N \leftarrow 1 - ['\text{present}']_G \rightarrow 2 \rightarrow ['\text{show}']_G$	$\equiv$ Some friends make the presentation of the show.
$['\text{friends}']_G \leftarrow 1 - ['\text{present}']_G \rightarrow 2 \rightarrow ['\text{show}']_G$	$\equiv$ The friends make the presentation of the show.

Figure 7: Expression of different Givenness distributions

⁵ Note that outside the MTT, Thematicity is two-partite in that it consists of Theme and Rheme only.

2.1.3.3. The Focalization Dimension

The *Focalization*-dimension identifies elements of an utterance that are logically important to the author himself (i.e., are in his focus of attention). A standard syntactic means to express focalization of an element is its fronting. Consider the expression of focalization of the semanteme ‘show’ in the first of the semantic structures in Figure 7:

9. *It is a show some friends are making a presentation of.*

2.1.3.4. The Foregrounding Dimension

The *Foregrounding*-dimension identifies elements of an utterance that are psychologically primary to the author (the corresponding elements are *Foregrounded*) and those that are unimportant to him (the corresponding elements are *Backgrounded*). Common syntactic means for foregrounding is raising and for backgrounding – appenditive parentheses. Cf., for illustration, Figure 8.

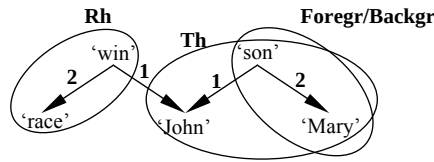


Figure 8: Foregrounded/Backgrounded elements in a SemS

If we foreground the fragment ‘son’ $\rightarrow$ ‘Mary’, the resulting sentence is 10; if we background it, the resulting sentence is 11:

10. *Mary’s son John won the race.*
 11. *John (who is Mary’s son) won the race.*

2.2. Grammar Modules of an MTM

With the definition of the representations at the different strata at hand, we can define the grammar modules of an MTM.

Definition: *Grammar Module* $M_{S_i, S_{i+1}}$

Let $\mathbf{R}_i$ and $\mathbf{R}_{i+1}$ be the multitudes of well-formed representations of the linguistic model at the strata S_i and S_{i+1} , respectively. Then, a (Grammar) Module $M_{S_i, S_{i+1}}$ consists of a set of elementary rules such that for any well-formed $R_i \in \mathbf{R}_i$ there is a subset of rules $M'_{S_i, S_{i+1}} \subseteq M_{S_i, S_{i+1}}$ such that

- (i) each rule from $M'_{S_i, S_{i+1}}$ establishes a correspondence between a minimal fragment of R_i and a minimal fragment of $R_{i+1} \in \mathbf{R}_{i+1}$
- (ii) each node and each arc of R_i is covered by at least and at most by one rule $M'_{S_i, S_{i+1}}$

For each pair of adjacent strata, a distinct grammar module has thus to be defined. The rules in the grammar modules are of a standard format: ‘ $L \Leftrightarrow R \mid \text{Conditions}$ ’, i.e., they contain a lefthand side representation $R_i \in \mathbf{R}_i$, a righthand side representation $R_{i+1} \in$

R_{i+j} and a number of (optional) conditions that restrict the equivalence. Consider five sample rules from different grammar modules in Figure 9. Rules 1 and 2 are samples from the Sem $\Leftrightarrow$ DSynt grammar module. Units in single quotes stand for semantemes; names beginning with a '?' are variables; 'Lex(α)' stands for the lexeme corresponding to the semanteme α . In Rule 1, the semanteme 'intensive' is mapped onto the LF Magn (of the lexeme that corresponds to the semanteme bound to the variable ?Xs), and the semanteme bound to ?Xs is mapped onto its lexeme. The predicate-argument relation between 'intensive' and ?Xs is mapped onto the DSynt-relation **ATTR**. Note also that the orientation of both relations is opposite. The rule applies only if the lexeme expressing the semanteme bound to ?Xs possesses a Magn-value. Rule 2 maps the semantic relation **1** onto the deep-syntactic relation **I** – if Lex(?Xs) has in its Government Pattern (GP) a first DSynt-actant, and, in case Lex(?Xs) is a verb, it is finite and no passive construction is foreseen.⁶

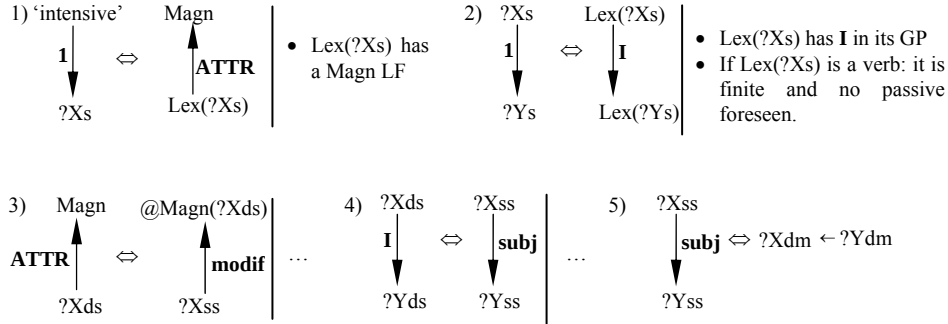


Figure 9: Sample grammar rules from different grammar modules

Rules 3 and 4 are samples of the DSynt $\Leftrightarrow$ SSynt module. In Rule 3, the Magn-LF label corresponds to the value of Magn applied to the lexeme bound to ?Xds: '@LF(λ)' stands for the application of the LF to the lexeme λ and use of the value lexeme.⁷ The deep lexeme ?Xds corresponds to the surface lexeme ?Xss, and the DSynt-relation **ATTR** corresponds to the SSynt-relation **modif**(ier). We dispense with the outline of the conditions for the application of this equivalence. Rule 4 exemplifies the correspondence between the DSynt-relation **I** and the SSynt-relation **subj**(jective) – including the correspondence between their arguments (again, certain conditions that we do not indicate here must be fulfilled).

Rule 5 defines the correspondence between the subjective relation and the precedence relation between the subject and its verbal governor – meaning that the subject comes linearly before the verb. Obviously, this rule is language-specific since in free word order languages, the order between all sentence elements is largely determined by the communicative structure.

As must have become clear, the rules imply correspondence links between nodes of the left hand side and those of the right hand side – although these usually are not made explicit. Thus, in the rules above, 'intensive' corresponds to Magn, '?Xs'

⁶ Note that the choice of the passive is due to the communicative structure at the semantic side and is handled by another rule.

⁷ For the interpretation of LFs as "deep lexemes" and as functions, see [30].

corresponds to ‘Lex(?Xs)’, etc. As we will note in Section 4, in a formal model for text generation, these links need to be made explicit.

2.3. The lexica in an MTM

Rich lexica play a key role in an MTM. These lexica are known as *Explanatory Combinatorial Dictionaries*, ECDs [21]. Each lexical entry in an ECD consists of a certain number of zones. The following three zones are of special relevance in the context of text generation: 1. the lexical definition of the LU in question, 2. the government pattern of the LU, 3. the lexical functions that apply to this LU.

The lexical definition presents the meaning of the LU. It can be more or less decomposed. Consider a possible definition of Eng. ADMIRATION:

Emotional attitude of X, favorable with respect to Y because of the action, state or property Z of Y, which X believes extraordinary.

X, Y, and Z are the semantic actants implied in the definition; in terms of semantic roles, X is the Experiencer, Y is the Object (of the attitude), and Z is the Cause (of the attitude).

The government pattern (GP) of an LU specifies: (i) this LU’s syntactic and semantic valency, (ii) the projection between the semantic and syntactic valencies, (iii) the subcategorization patterns of the LU. Semantic valency specifies the semantic actant slots determined by the meaning of the LU; syntactic valency does so with respect to syntactic actant slots. For the majority of the LUs, semantic and syntactic valencies are projected one-to-one onto each other – as in the case of ADMIRATION.⁸ Table 1 shows a standard GP-table for Fr. ADMIRATION ‘admiration’. ‘N’ stands for noun or nominal phrase, ‘A_{poss}’ for possessive adjective, and ‘A’ for adjective.

Table 1: Government Pattern for Fr. ADMIRATION

X = 1 = I	Y = 2 = II	Z = 3 = III
1. de ‘of’ N 2. A _{poss} 3. A	4. de ‘of’ N 5. pour ‘for’ N 6. devant ‘in front’ N 7. envers ‘towards’ N	8. pour ‘for’ A _{poss} + N

The LFs that apply to an LU are listed together with their values. Cf. some LFs of Fr. ADMIRATION and of Eng. ADMIRATION in Table 2.⁹

Table 2: Sample LFs for Fr. ADMIRATION and Eng. ADMIRATION

Able ₁	<i>sujet</i> ‘subject’, <i>enclin</i> ‘tend’, <i>porté à</i> [ART ~] ‘carried to’	<i>open</i> [to ~]
Able ₂	<i>digne</i> ‘worth’	<i>be worthy</i> [of ~]
Ver ₁	<i>justifiée</i> ‘justified’, <i>fondée</i> ‘funded’, <i>sincère</i> ‘sincere’	<i>justified</i> , <i>well-founded</i> , <i>sincere</i> , ...
AntiVer ₁	<i>injustifiée</i> ‘unjustified’, <i>de commande</i> ‘of the order’	<i>grudging</i>
Oper ₁	<i>éprouver</i> ‘feel’, <i>ressentir</i> ‘sense’, <i>avoir</i> ‘have’, ...	<i>feel</i> [~], <i>have</i> [~]
IncepOper ₁	<i>tomber</i> [en ~] ‘fall in’	<i>fall</i> [in ~]

⁸ However, this is not always the case; cf. e.g. FLY, where Z=3 (the Instrument) is usually not expressed by a syntactic actant: *He flew from Barcelona to Montreal*.

⁹ The translations of the French LF-values are literal.

FinOper ₁	<i>perdre</i> [ART ~] ‘lose’	<i>lose</i>
Oper ₂	<i>attirer</i> ‘attract’, ...	<i>compel, draw</i>
ContFunc ₀	<i>durer</i> ‘last’	<i>last</i>
FinFunc ₀	<i>s’éteindre</i> ‘extinguish’, <i>s’évanouir</i> ‘vanish’, ...	<i>vanish</i>

3. The Basics of Text Generation

Text generation is the production of written material from a formal representation. The formal representation which serves as input can extremely vary: some text generators start from a surface-oriented syntactic structure, while others start from extra-linguistic numerical data series. Samples of MTT-generators are available for both extremes. Consider, e.g., [31] for a “shallow” syntactic generator and [32, 33] for “deep” generators. In order to cover the whole range of MTM-strata, we focus on the latter.

From the perspective of deep generation, text generation involves the following major tasks:

- (1) *Content selection*: selection of the content that is to be rendered in a textual format from a given data or knowledge base.
- (2) *Discourse structure planning*: construction of a text plan that determines in which order the discourse units are to be presented and how they are linked in order to ensure coherence.
- (3) *Sentence planning*, which in its turn consists of several demanding subtasks:
 - a. *Aggregation*: fusion of structures or content to avoid reduplication,
 - b. *Sentence packaging*: determination of sentence boundaries,
 - c. *Sentence structure determination*: selection of the overall syntactic structure of sentences,
 - d. *Lexicalization*: Mapping of semantic units encountered in the content (semantic) structures onto lexical units.
- (4) *Surface realization*: Spelling out of the morphological features associated with the lexical units.

Let us briefly introduce each of these tasks in order to be able to discuss then in Section 4 how an MTM is used for text generation.

3.1. Content Selection

The task of content selection assumes that not all content that is available to the generator is to be communicated to the reader. Strictly speaking, content selection is not a text generation task in the sense text generation has been defined above. Rather, it implies the assessment of the relevance of the given data or knowledge sources with respect to the addressee’s profile, the context of the situation and the general state of affairs as judged by an expert of the field. It can be thus realized as an “expert system shell” prior to rendering the relevant data/knowledge structures into a textual format by a text generator, into a table format by a table generator or into a graphic form by a graphic generator.¹⁰

Nonetheless, content selection is traditionally considered a part of text generation. Often, it is combined with discourse planning into a more complex task, namely *text*

¹⁰ The choice of the most appropriate mode for rendering the relevant information to the reader is another task in *multimodal* text generation.

(or *macro*) *planning*. How appropriate this combination is depends on the genre of the text under generation. For narration, it is natural to select the content as the story goes on; for reports on numerical time series, the content that is to be communicated is better chosen beforehand using expert knowledge. For instance, for air quality bulletins, the information on threshold exceedance, on highest and/or lowest concentrations of air pollutant substances, on remarkably sharp rises and falls of the concentrations due to changing meteorological conditions, etc. is predestined for communication [34]. Thus, if, as shown in Figure 10, the air quality index reaches at 16:00 the highest mark on a 6 grade scale, both the mark and its rating should be transmitted to the addressee.

time	11:00	...	16:00	AQ-index	1	2	3	...	6
AQ-index	3	...	6	rating	very good	good	fair	...	very bad

Figure 10: Numeric time series in the air quality domain

One of the formats for the codification of the selected content are the *Conceptual Graphs*, CG [35]. CGs are semantic networks with *concepts* as nodes and *semantic roles* (such as OBJ(ect), VAL(ue), R(e)S(u)LT, ATTR(tribute)) between the concepts as arcs. The concept nodes can be typed. Consider Figure 11 for the codification of the information highlighted in Figure 10 in grey as a conceptual graph.

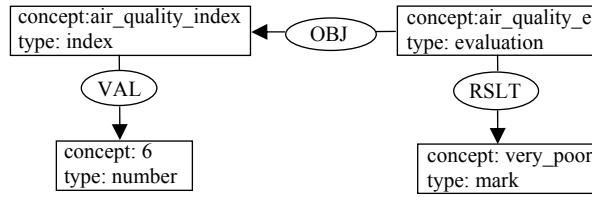


Figure 11: Codification of content in terms of Conceptual Graphs

3.2. Discourse structure planning

Two major strategies for discourse structure planning are known: (i) schema-driven planning [36] and (ii) Rhetorical Structure Theory (RST)-driven planning [37]. Schemata, or *scripts* [38], encode standard patterns of the discourse structure in terms of predefined sequences of rhetorical predicate constructions. Their use is convenient in the case of a static discourse and content structure – as, e.g., in weather report generation [32]: as a rule, the past, current and future atmospheric and meteorological conditions are always presented in the same sequence such that one general discourse structure (or schema) can be defined and used for any input.¹¹ RST-driven planning implies a dynamic construction of a discourse tree or graph in which between the “discourse spans” coherence relations of the type ELABORATION, INTERPRETATION, CAUSE, JUSTIFICATION, etc. hold [39, 40]. For instance, between the two statements from Figure 11 ‘air_quality_index is 6’ and ‘air_quality_evaluation results in “very poor”’, the INTERPRETATION-relation holds;

¹¹ In practice, the discourse schemas are usually dynamic to a certain extent in that they can be adapted to the changing input data by changing the order of the appearance of some fragments or dropping them.

cf. Figure 12. Since INTERPRETATION is an asymmetric relation, it is directed. ‘N’ stands for governor or “nucleus”, and ‘S’ for dependent or “satellite”.

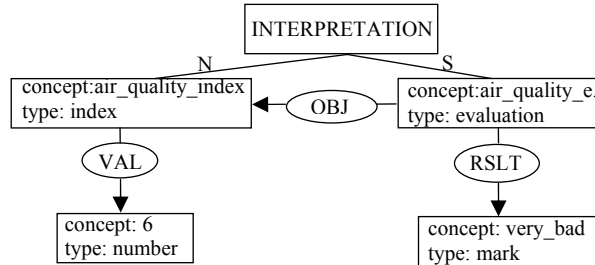


Figure 12: Illustration of the discourse relation INTERPRETATION

Schema-based and RST-based planning strategies can be combined [41, 42, and 43].

3.3. Sentence planning

Sentence (or *micro*) planning is certainly the most challenging task in text generation and still awaits its final solution. State of the art text generators – including MTT-based generators – address the individual subtasks of sentence planning in isolation, some of them rather *ad hoc* and shallow (as, e.g., sentence packaging and sentence structure determination), others in more depth (as, e.g., aggregation and lexicalization, and especially such aspects of lexicalization as referring expression generation).

3.3.1. Aggregation

As mentioned above, aggregation deals with the fusion of linguistic or knowledge structures to avoid reduplication of information and thus to achieve a higher degree of text conciseness; see, among others [44, 45]. A reduplication of information can be of contextual, conceptual, lexico-semantic, or syntactic nature; cf. the following examples:¹²

12. a. *Last year, John spent his summer holidays in Barcelona.*
 b. *In the summer of 2007, Mary spent her holidays in the capital of Catalonia.*
 c. *In the summer of 2007, James spent his leisure time in Barcelona.*
13. *Last year, John, Mary and James spent their summer holidays in Catalonia's capital Barcelona.*

Sentence 13 contains the information from 12a-c in a concise form. To be able to fuse ‘last year’ and ‘2007’, we need to know that the year of reference is 2008 – which is contextual information. That summer holidays are during the summer and that Barcelona is the capital of Catalonia is extra-linguistic, i.e., conceptual information. The information that ‘leisure time’ and ‘vacations’ mean (roughly) the same is of lexico-semantic nature and the realization of the possessive construction *in Catalonia's capital Barcelona* instead of the repetition of *in Barcelona* and *in the capital of Catalonia* is of syntactic nature.

¹² To save some space, we give here examples in terms of actual sentences rather than in terms of the corresponding structures.

Each type of aggregation can be handled by aggregation rules at the corresponding level of linguistic representation.

3.3.2. Sentence packaging

The central task of sentence packaging is to partition a given semantic structure into sentences – without, however, having necessarily determined the structure of these sentences. This task is highly language-specific. For instance, English favours much shorter sentences than German and Russian. In current text generators, sentence packaging is solved more or less *ad hoc*. Most often, the conceptual structures (or, after discourse planning, the discourse spans) already implicitly predetermine the division of the content into clauses and subsequently sentences. For illustration, Figure 13 shows the semantic structure corresponding to the conceptual structure of Figure 12 and its packaging into two sentences.

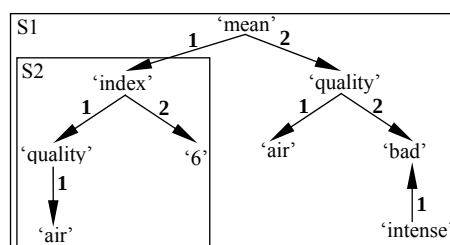


Figure 13: Sentence packaging in a semantic structure

3.3.3. Sentence structure determination

Sentence structure determination can be performed together with sentence packaging or after it. In any case, the structure of a sentence is decisively influenced by the communicative structure. As illustrated in Subsection 2.1.2 above, dimensions such as Thematicity, Focalization, etc. require specified elements to be realized as syntactic subject or object, to be fronted, raised, etc. Essential is also the information on the communicative dominance of an element in a given sentence structure. Thus, the dominance of the semanteme 'mean' in the S1 package of Figure 13 will lead to a sentence like *S2 means that the air quality is very poor*, while the dominance of the semanteme 'quality' will lead to a sentence like *Given S2, the quality of the air is very poor*.

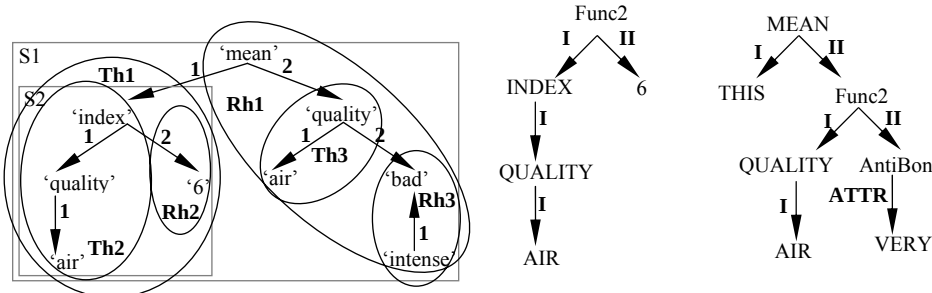


Figure 14: Interdependency between CommSemS and DSyntS

Some works, in fact, treat the role of the communicative structure in generation; cf., among others, [46, 47, 48]. However, the majority of the generators foresee a default sentence structure that mirrors the semantic or conceptual structure. To take our sample semantic structure from Figure 13 further, consider in Figure 14 (previous page) one of its possible communicative structures and the DSynt-structures that correspond to this communicative structure. The realization of ‘index’ $-1 \rightarrow$ ‘quality’ $-1 \rightarrow$ ‘air’ as Theme and ‘6’ as Rheme in S2, leads to a DSyntS with the Func_2 -LF as DSynt-head such that the NP headed by INDEX is the first DSynt-actant and ‘6’ the second DSynt-actant. At the SSynt-stratum, the default realization of **I** is **subjective** and of **II** it is **dir.objective**. In S1, S2 is as a whole Theme, so the most adequate realization is the sentential reference THIS as DSyntA **I**. With Th3 and Rh3 embedded in Rh1, the corresponding DSyntS must foresee a clause realization of Th3+Rh3; Func_2 as governor fulfils the constraints.

3.3.4. Lexicalization

As mentioned above, lexicalization has been one of the most popular research topics in sentence planning or even in text generation in general. It is a two-partite task: 1. lexicalization proper, i.e., finding adequate names for the individual semantic units or configurations of semantic units, and 2. choice of referring expressions in the case of a multiple occurrence of a name in a sequence of sentences in order to ensure cohesion. In some frameworks – including MTM – both subtasks are performed at different stages of generation.

The lexicalization algorithms proposed in the literature differ widely [49]. Some assume a one-to-one association of words with semantic units such that lexicalization is realized as a search in the semantic resources – be they an ontology or a conceptual net [50, 51]; some match fragments of the semantic structure with definitions of lexical units [52]; and others do first a direct dictionary look-up and explore then the best lexicalization among available (quasi-synonymous) options [48, 53]. Note that in Figure 14, lexicalization is done in parallel with the sentence structure determination, i.e., DSyntS already contains deep lexemes.

For work on the choice of referring expressions as a separate task, see, e.g., [54, 55, 56, and 57].

3.4. Surface generation

Surface generation is the less demanding and best mastered part of text generation. Its scope may vary in that it may cover generation starting from ordered (surface-) syntactic structures or simply the instantiation of the morphological features associated with the individual lexemes. But usually, it does not go beyond a straightforward realization of predetermined surface structures.

3.5. Summary: The linguistic framework and the requirements of text generation

Various linguistic theories have been used as theoretical foundation of text or sentence generators. However, the tasks of generation require some features that are easier and more naturally provided by some theories than by others. Let us summarize the most important of these features and assess what they imply:

- As a rule, generation is required to start from an abstract representation or even from numerical data series
 - ⇒ given that abstract (conceptual or semantic) structures are predicate-argument structures, the use of a dependency theory is more straightforward.
- Involves planning of text structures
 - ⇒ requires supra-sentence modelling instruments
- Involves planning of sentence structures
 - ⇒ requires the availability of the communicative structure
- Must be able to vary and use idiosyncratic expressions and word combinations
 - ⇒ requires rich lexica that are fully integrated into the generation procedure
- Experience has shown that it is most adequate to treat the different tasks of generation separately
 - ⇒ favours a multistratal model
- Should be formally verifiable
 - ⇒ requires a formal framework beneath the generator
- Should be easily maintainable
 - ⇒ favours a modular organization

In the next section, we discuss how an MTM can be used for generation and to what extent it meets the above requirements.

4. Text Generation Using MTMs

MTT is one of the most popular linguistic theories in text generation because it possesses all of the required features – with the exception of the notion of discourse structure and text planning mechanisms, which need to be interfaced with an MTM-based generator kernel. Before we discuss how this is done, we need to reinterpret the MTM in the sense of a generator: as pointed out above, MTM is an equivalence or correspondence model, while generation presupposes a transition model. More precisely, we can view generation in the sense of an MTM as a sequence of transitions between representations of adjacent strata – starting from the semantic stratum, up to the surface-morphological stratum. Let us have a more formal look at the realization of this transition model.

4.1. Formal realization of the MTM

The following characteristics of the MTM when viewed as transition model are decisive for its realization; see [8]:

- transitions take place between two adjacent levels of representation; no intermediate representations containing fragments of representations from both levels are admitted;
- the correspondence between chunks of representations is static;
- access is available to the source structure and the target structure of the transition as well as to the context of both structures;
- the transition rules are (at least theoretically) bidirectional and can be used in parallel.

These characteristics suggest the use of *finite state transducers* [58], only that in the case of an MTM and text generation, we have to deal with *graph transducers* rather than with *string transducers*.

4.1.1. MTM-grammar modules for generation

In order to be able to use the MTM-grammars presented above for generation, the following three actions must be taken:

- (i) the rules must be made more exhaustive, explicit and formal;
- (ii) an algorithm that applies the rules of a given module to an input structure of the stratum S_i and transduces it into an output structure of the stratum S_{i+1} must be defined and implemented;
- (iii) the format of the dictionaries and the access of the dictionaries by the transduction algorithm must be formalized.

In what follows, we present these three actions as realized in the MATE workbench [59, 60].

4.1.1.1. Grammar rule format in MATE

A grammar $G_{i \rightarrow i+1}$ to map a structure of S_i to a structure of S_{i+1} in MATE consists of a set of minimal grammar rules of the following general format; see [60, 39ff) for details:¹³

left side (ls):	$\langle g_i \rangle$
right side (rs):	$\langle g_{i+1} \rangle$
right context (rc):	$\langle g'_{i+1} \rangle$
conditions (cd):	$\langle \text{Boolean expression over } \mathcal{D}, S_i, \text{ and } S_{i+1} \rangle$
correspondences (cr):	$\{n_{i,j} \Leftrightarrow n_{i+1,k}\}$

with g_i being a graph defined over the node and arc alphabets of S_i , g_{i+1} and g'_{i+1} graphs defined over the node and arc alphabets of S_{i+1} ; $\mathcal{D}$ being the dictionaries, $n_{i,j}$ being a node $\in g_i$, and $n_{i+1,k}$ being a node $\in g_{i+1}$. The statement ' $n_{i,j} \Leftrightarrow n_{i+1,k}$ ' establishes a link between the corresponding nodes in g_i and g_{i+1} in order to ensure that (i) information can be propagated from one node to another node across strata; (ii) isolated fragments of the target structure introduced upon the application of the rule can be unified to one connected well-formed structure. A rule is applied to an input structure defined over the alphabets of S_i if the specified conditions are fulfilled and if an isomorphic image of g'_{i+1} (the right context) has been identified in the target structure. As indicated, the conditions may be defined over all dictionaries and both strata S_i and S_{i+1} .

The left side 'ls' of a rule in $G_{i \rightarrow i+1}$ consists either of an elementary linguistically meaningful graph defined over the alphabets of S_i or a graph that is transduced to an elementary linguistically meaningful graph defined at the right side 'rs' over the alphabets in S_{i+1} . As a rule, an elementary meaningful graph consists either of a single node (a linguistic name) or a single arc (a linguistic relation) – although sometimes bigger structures are required. For illustration, consider sample rules for the first four types of transductions involved in MTT-based generation from a conceptual structure: a Con-Sem rule, a Sem-DSynt rule, a DSynt-SSynt rule, and a SSynt-DMorph rule.

¹³ See also [10] for a more detailed exposition.

Rule 1 (a Cons-Sem rule):

```

ls: ?Xcon {PTIM->?T{con= "tomorrow"}}
rc: ?Xsem{tense=FUT}
cr: Xcon⇔?Xsem

```

Rule 1 maps the conceptual time relation PTIM between the concept denoted by the variable ?Xcon and the “universal” concept TOMORROW onto the tense feature ‘FUT’ of the semanteme denoted by the variable ?Xsem. Note that ?Xsem is specified in the right context slot, which means that the semanteme bound to ?Xsem is assumed to have already been introduced into the target structure by another rule.

Rule 2 (a Sem-DSynt rule):

```

ls: ?Xsem {?r->?Ysem}
rs: ?Xds{I->?Yds}
rc: ?Xds
cr: Xsem⇔?Xds; ?Ysem⇔?Yds
cd: lexicon::(?Xds.lex).(gp).(r)=I

```

Rule 2 maps the semantic relation (denoted by the variable ‘r’) of the semanteme bound to ?Xsem onto the DSynt-relation I of the lexical unit L bound to ?Xds. The conditions ensure that this is in accordance with the government pattern of L. The node ?Xds must already be present in the target structure. The semantic node bound to ?Xsem corresponds to the deep-syntactic node bound to ?Xds and the node bound to ?Ysem corresponds to the node bound to ?Yds.

Rule 3 (a DSynt-SSynt rule):

```

ls: ?Xds {dpos=V; finiteness=FIN; mood=IND; tense=FUT}
rs: ?Xss{ slex=will
          dpos=lexicon::(will).dpos
          spos=lexicon::will).spos
          tense=PRES
          aux_completive->?Xss{finiteness=INF}
rc: ?Xss
cr: Xds⇔?Yss; ?Xds⇔?Xss
cd: lexicon::(id).(iso)=ENG

```

Rule 3 introduces for an English verbal lexeme (referred to by ?Xds) that carries in the DSyntS the grammemes FIN, IND, and FUT, the auxiliary WILL. WILL inherits from ?Xds via the correspondence link the grammemes of finiteness and mood, but not of tense – which is PRES. Note that in this case, one DSynt-node corresponds to two SSynt-nodes.

Rule 4 (a SSynt-DMorph rule):

```

ls: ?Xss {dpos=V
          subj->?Yss
          ?r->?Zss}
rs: ?Ytp{b->rc: ?Ztp}
cr: ?Yss⇔?Ytp; ?Zss⇔?Ztp
cd: not r=circumstantial

```

Rule 4 defines the relative ordering between the subject (bound to ?Yss) of a verbal lexeme (referred to by ?Xss) and any other dependent of the verb (?Zss): the subject goes before. The circumstantial elements are excluded since they may come before the subject.

4.1.1.2. Transduction

The algorithm adopted by MATE's transduction engine to map a source structure G_s defined over the alphabets of the stratum S_i onto a target structure of the stratum S_{i+1} using the grammar $G_{i \rightarrow i+1}$ can be sketched as follows:

Given, a graph system $(G_s, G_{i \rightarrow i+1}, \mathbf{G}_t)$ with G_s as the source graph, $G_{i \rightarrow i+1}$ as the rule grammar, $g \in G_{i \rightarrow i+1}$, $g = (ls, rs, rc, cd, cr)$ and $\mathbf{G}_t$ as the set of target graphs, map G_s onto $G_i \in \mathbf{G}_t$ performing the following steps:

1. Binding: Identify all isomorphic images of $ls \in g$ ($\forall g \in G_{i \rightarrow i+1}$) in G_s and associate their elements with the corresponding elements of ls .
2. Evaluation of conditions: Evaluate the conditions for each bound image of ls .
3. Clustering: Build groups of rules, $\mathbf{CL}$, with a group (or, cluster) $CL \in \mathbf{CL}$ containing all applicable rules that do not contradict each other and mapping G_s in its entirety.
4. Application: Create the images of $rs \in g$ ($\forall g \in CL, \forall CL \in \mathbf{CL}$)
5. Unification: Unify the images of all rs that belong to the same cluster CL .

The goal of the algorithm is thus to create for a given G_s all possible well-formed structures $\mathbf{G}_t$ that correspond to G_s in accordance with $G_{i \rightarrow i+1}$, avoiding incomplete G_s , i.e., G_s that do not entirely cover G_s and structures that are corrupted because several rules with an overlapping left-hand side graph have been applied.

4.1.2. Generation dictionaries

The propositional structures at each stratum are characterized by a distinct node and arc alphabets. During the transitions, elements or configurations of elements of this alphabet are put into correspondence to (configurations of) elements of the alphabet at the adjacent stratum. As a result, generation dictionaries contain two types of information: (i) information concerning the elements of the alphabet of a given stratum, (ii) information concerning the correspondence between elements of alphabets of adjacent strata. Since MTT is a lexicalist (i.e., node alphabet-oriented) theory, the dictionaries concern node alphabet elements.

At least two dictionaries are available: 1. a "semantic dictionary", which contains the information required with respect to semantic units (especially the information how semantic units are mapped onto lexical units), and 2. a "lexical dictionary" with the information on lexical units. Other dictionaries, such as a morphological dictionary can also be introduced. For generation from extra-linguistic abstract (conceptual) representations, furthermore, an additional, conceptual, dictionary is required. In what follows, we discuss first such a conceptual dictionary and then a semantic and a lexical dictionary.

4.1.2.1. Conceptual dictionary

The conceptual dictionary contains, first of all, the type and argument structure of each concept and the concept-semantic mapping information; cf., for illustration, the entry for the concept CONCENTRATION.

```
concentration: property_attribute {  
  sem = 'concentration'  
  MATR = {relation = 1 target = referent}  
  VAL = {relation = 2 target = referent}  
  ATTR = {relation = 1 source = referent}}
```

The concept CONCENTRATION has two argument slots: something that has a concentration (referred to as MATR in accordance with [35]) and a value (referred to as VAL), i.e., an absolute concentration figure. The concept may also be modified by a qualitative characterization of the concentration (“high”, “low”, etc.), referred to as ATTR. The corresponding semantic ‘concentration’ takes MATR as its first semantic argument (indicated by the “relation=1” parameter embedded in MATR’s value) and VAL as its second. The attributes “target=referent” and “source=referent” indicate the direction of the semantic relation (for MATR and VAL, the semantic predicate is ‘concentration’, which takes MATR’s and VAL’s corresponding semantemes as its arguments, while ATTR’s semantic correspondent is a predicate taking ‘concentration’ as its argument).

4.1.2.2. Semantic dictionary

The semantic dictionary gives, for each semantic, all its possible lexicalisations. For instance, for the meaning ‘cause’ it lists the LUs CAUSE[V], CAUSE[N], RESULT[V], RESULT[N], DUE, BECAUSE, CONSEQUENCE, etc. Note that we do not consider at this stage the valency of the LUs. Thus, it does not matter that ‘X causes Y’ means that ‘Y results from X’; what interests us here is only that these two lexemes can both be used to denote the same situation, regardless of the communicative orientation (in a more comprehensive dictionary, the criteria for the choice of a specific lexicalization are specified). Cf., for illustration, the entry for the semantic ‘concentration’ as specified in the semantic dictionary:

```
concentration {  
  label = parameter  
  lex = concentration}
```

The “label=parameter” attribute-value pair specifies the semantic type of a semantic. We can also specify the semantic type of the arguments of a predicate; for instance, adding the attribute “1=substance” would force the first semantic argument of ‘concentration’ to be of type “substance”.

4.1.2.3. Lexical dictionary

The lexical dictionary is the richest of the set of three generation dictionaries. It contains, for each LU, information on its part of speech, minimal sub-categorization information and the LFs that are applicable to the LU. Consider, for illustration, the entry for CONCENTRATION:

```

concentration {
// Grammatical characteristics:
    dpos=N // deep part of speech is N(oun)
    spos=common_noun // surface part of speech is common noun
// Government pattern (subcategorization):
    gp={
// Sem-DSynt valency projection:
        1=I // first semantic actant corresponds to the first deep-syntactic actant
        2=II // second semantic actant corresponds to the second deep-syntactic actant
// First syntactic actant can be realized as "ozone concentration":
        I={
            dpos=N // actant is a noun
            rel=compound // linked with compound relation
            det=no // takes no determiner
        }
// First syntactic actant can be realized as "concentration of ozone":
        I = {
            dpos=N // actant is a noun
            rel=noun_completive // linked with noun_completive relation
            prep=of // takes preposition "of"
            det=no // takes no determiner
        }
// Second syntactic actant can be realized as "concentration of 180 µg/m3":
        II={
            dpos=Num // actant is a number
            rel=noun_completive // linked with noun_completive relation
            prep=of // takes preposition "of"
        }
    }
// Lexical functions:
    Magn = high
    AntiMagn = low
    Adv1 = in // "(we found) ozone in a concentration (of 180 µg/m3)"
    Func2 = be // "the concentration (of ozone) is 180 µg/m3"
    Oper1 = have // "ozone has a concentration (of 180 µg/m3)"
    IncepFunc2=reach // "the concentration (of ozone) reached 180 µg/m3"
    IncepOper1=reach // "ozone will reach a concentration (of 180 µg/m3)"
}

```

It is convenient to use two levels of granularity for the part of speech, referred to as “deep” and as “surface” part of speech (‘dpos’ and ‘spos’, respectively). This allows for quick reference to a whole family of parts of speech in grammar rules (for example, “N” refers to any proper noun, common noun, or pronoun). All specific grammatical characteristics of an LU are described as feature-value pairs (for example, its gender or its ability to take or not plural, definiteness, a certain tense, etc.). The sub-categorization must contain the projection of the semantic to the syntactic valency and all possible ways of syntactically connecting the LU with its dependents. Governed prepositions must be indicated here, as well as case assignment if it exists in the language being described. The part of speech of the dependents can be restricted (for instance, the first actant of CONCENTRATION must be a noun, while its second must be a number).

As already outlined above, LFs are an efficient means to refer to recurrent semantic and syntactic patterns of restricted lexical co-occurrence and lexico-semantic derivation. In the example above, Magn points to an LU which is a syntactic modifier

and has a meaning of intensification (AntiMagn is its antonym). The LF Func₂ refers to a semantically emptied verb that takes the keyword (CONCENTRATION) as its subject and the keyword's second semantic actant as its object. The LF IncepOper₁ points to a verb meaning roughly 'start' which takes the keyword as its object and the first actant of the keyword as its subject.

In general, we can state that the more information on restricted lexical co-occurrence the lexical dictionary contains the more natural and idiomatic the generated text will be.

4.2. Applying MTM in Generation

Now that we have a more formal outline of an MTM, let us assess it from the viewpoint of the generation tasks talked about in Section 3.

4.2.1. Text planning in an MTM-based generator

The text planning task is covered by a module that operates on an extra-linguistic representation outside the linguistic generator kernel and that provides as output a *text plan*. The text plan serves as input to the linguistic generator. Given that the most abstract stratum this generator starts from is either the semantic or the conceptual stratum, the text plan is required to be expressed as/mapped on a semantic or on a conceptual graph representation. This does not mean, of course, that the planning module is required to operate on a semantic/conceptual representation.

In practice, both schema-based and RST-based planning mechanisms have been used in MTM-based generation; cf. [61, 62] for the first and [63, 42] for the second. Figure 15 displays a sample text plan as produced by the planning module described in [42].

```
<span id="UN65932" relation="topic-focus-elaboration" modes="text">
  <span id="UN65938" node="nucleus">
    <infounit id="UN65941">
      <topic id="UN65943" section="current" type="aq" subtypes="secondary" alert="false"/>
    </infounit> </span>
  <span id="UN65951" relation="negative-justification" node="satellite">
    <span id="UN65956" node="nucleus">
      <infounit id="UN65959" order="1.2">
        <rating id="UN65961" ref="SN65568" label="1/6" possessor="aq" region="Catalunya"/>
      </infounit> </span>
    <span id="UN65973" relation="list" node="satellite">
      <span id="UN65977" node="nucleus">
        <infounit id="UN65980" order="1.4.1">
          <concentration id="UN65982" possessor="O3"/>
        </infounit> </span>
      <span id="UN65991" node="nucleus">
        <infounit id="UN65994" order="1.4.2">
          <concentration id="UN65996" possessor="SO2"/>
        </infounit> </span>
    <span id="UN66005" node="nucleus">
      <infounit id="UN66008" order="1.4.3">
        <concentration id="UN66010" possessor="CO"/>
      </infounit> </span>
    </span> </span> </span>
```

Figure 15: Fragment of a sample text plan used as input to the linguistic generator

Such a text plan can be readily mapped onto a conceptual graph structure that is “understood” by the linguistic MTT-based generator.

4.2.2. Deriving the communicative structure

As we have seen in Section 3, the communicative structure is very important in MTT-based text generation. However, when we start from an abstract conceptual representation or from a numeric time series, CommS is not available. It must be derived in the course of generation. Some preliminary work on how this can be done is described in [64]. The central assumption underlying this work is that the communicative structure of a statement is determined by:

- (iv) domain communication knowledge concerning the context of the statement: each domain has its own way “to tell the story”,
- (v) the discourse structure relations in which the statement is involved: RST-like relations give clear hints with respect to the distribution of the communicative structure.

Based on this assumption, rules of the following kind are derived for generation from numeric time series:

- IF** 1. between $\text{DU}^{\text{Sta-}}$ and DU^{Sta} CONTRAST holds
 2. the contrasted elements are $v^- \in \text{Sta}^-$ and $v \in \text{Sta}$ of t wrt circumstantial c
 3. $t \in \text{Th}^{\text{Sta-}}$, $v^- \in \text{Rh}^{\text{Sta-}}$, $c \in \text{Rh}^{\text{Sta-}}$
THEN $\text{Th}^{\text{Sta}} \leftarrow t$; $\text{Th}^{\text{Sta}} \leftarrow c$, $\text{Rh}^{\text{Sta}} \leftarrow v$

(with ‘Sta’ as statement under construction, ‘Sta⁻’ as one of the preceding statements, ‘DU^{Sta}’ as the discourse unit containing the statement Sta, ‘t’ as token captured by the time series, ‘v’ as value of the token at a given time, ‘Th^{Sta}’ as Theme of the statement Sta and ‘Rh^{Sta}’ as Rheme of the statement Sta).

Using rules of this kind for the other primary dimensions such as Givenness, Focalization, and Foregrounding, a rich enough CommS can be derived to guide the subsequent generation.

4.2.3. Coping with sentence-planning tasks in an MTM-based generator

In the MT-transduction model as implemented for text generation, most of the sentence-planning tasks are realized during the Sem-DSynt transition; some aspects of lexicalization (such as anaphoric reference introduction) are naturally handled during the DSynt-SSynt transition. The surface-realization is done starting from the DSynt-SSynt transition through all remaining subsequent transitions.

As far as aggregation is concerned, the MTM covers lexico-semantic and syntactic aggregation; contextual and conceptual aggregation must be carried out outside (prior to) the linguistic generator proper – for instance, when mapping the conceptual structure to the semantic structure. Consider, for illustration, a sample rule that handles first actant aggregation for English.¹⁴

¹⁴ Note that lexico-semantic and syntactic aggregation can also be carried out at a single stratum, namely the DSynt-stratum. However, this would require a *structure rewriting model*, rather than a transduction model. The general usefulness of such a rewriting model is undisputable – for instance for paraphrasing or summarization exercises.

Rule 5:

```

ls: ?S{?X1sem
    ?X2sem}
rs: ?X1ds{COORD->?Ads{
    lex=AND
    II->?X2ds{}}}
rc: ?Xds; ?Yds
cr: X1sem⇔?X1ds; ?X2sem⇔?X2ds
cd: ?S{?X1sem{1->?Ysem}
    ?X2sem{1->?Ysem}}

```

This rule applies in case the same first semantic actant occurs twice with different governors. The result of the rule at the DSynt-side is a coordination of the DSynt-realization of these governors.

In Subsection 3.3, sentence packaging can be interpreted as taking place either at the semantic stratum as a structure rewriting procedure, or during the Con–Sem transition. With the semantic stratum as the most abstract linguistic stratum, MTT foresees the first option. However, when a generator starts from a conceptual structure, the second option is more practical and more natural. So far, MTT-based generators are not an exception when compared to other state-of-the-art generators with respect to sentence packaging: they tend to adopt a rather *ad hoc* strategy of sentence packaging: a main verb with its argument structure forms by default a sentence; the sentence can be extended by relative clauses modifying the arguments of the main verb. We dispense here with giving rule examples for sentence packaging.

The transition model in an MTM allows for lexicalization strategies of various complexities. A comprehensive semantic dictionary contains all possible lexicalizations of each semanteme; see Subsection 4.1.2.2. The criteria for the selection of one specific lexicalization are to be specified in terms of attribute-value pairs in the dictionary.

The task of sentence structure determination is carried out during the Sem-DSynt and DSynt-SSynt transitions. As pointed out in Subsection 3.3.3, it is largely guided by the communicative structure.

5. Principles for Compiling Generation Resources

One key to the solution of the problem of the shortage of resources for low- and middle-density languages is the maximal reuse of the resources available for other languages. Existing resources can be reused in two different ways: (a) by *resource porting* [65, 66] and (b) by *resource sharing* [67, 68, 69, 10]. *Porting* implies the adaptation (by copy and modification) of the resources developed for language $\mathcal{L}_1$ to language $\mathcal{L}_2$, while *sharing* implies the extraction of resources shared by all languages considered and their common maintenance. Intuitively, porting seems more suitable for largely diverging $\mathcal{L}_1$ and $\mathcal{L}_2$ and sharing for typologically similar languages.

In our work, we focused so far on resource sharing for grammatical resources and resource porting for lexical resources [10]. For effective resource sharing, it is crucial that already the design of the organization of the resources supports resource (re)use across languages. We adopt the following strategy: (i) extract recurrent rule patterns across languages and factorize them out into a “meta-grammar”, (ii) modularize (or packaging) language-specific rules, (iii) shift the bulk of the grammarian’s work to the

lexicon, (iv) generalize recurrent lexical patterns and introduce an inheritance mechanism. This strategy has been successfully applied to the development of resources for Catalan, English, French, Polish, Portuguese and Spanish.

5.1. Extracting recurrent rule patterns across languages

In general, a contrastive analysis of multilingual grammatical resources quickly shows that many of the rules are shared by a subset of the languages, or even by all languages under consideration. For instance, Rule 1 in 4.1.1.1, maps the concept ‘TOMORROW’ onto the semanteme ‘future’. In subsequent transduction stages towards the surface, ‘future’ is realized by language-specific morphological or lexical means, but at the semantic stratum, it merely encodes the meaning ‘one day later than now’ – which is not specific to any language. Similarly, Rule 2 makes no reference to any specific lexical unit (LU), nor does it refer to any language-specific grammatical relation (semantic and deep-syntactic relations are universal by definition). Rather, it implements the government pattern of any given LU with respect to its DSyntA **I** slot. Analogous rules are available for all DSyntA slots (**II**, **III**, ...). Rules of this kind are good candidates to universality – although no absolute statements with respect to the universality of any grammar rules can be made at the current state of research in language engineering.

In contrast, Rule 3 is only valid for English. It refers to a specific lexeme (WILL), and it even explicitly requires that the ISO identification code of the language be “ENG”.

Between these two extremes, rules are encountered that apply to a set of languages – a family or any *ad hoc* set. For instance, Rule 4 is an example of a (word order) rule which by default holds for all languages we considered so far (even for Polish as a free-order word language shows a strong preference for ordering the grammatical subject before the verb it is governed), but is certainly not universal. Furthermore, in English, the determiner in most cases agrees with its governing noun only in number, while, for instance, in Romance and Slavic (in case a determiner other than article is available) languages, it agrees in number and gender (in Slavic languages, also in case). Therefore, we can specify one rule for all these languages – excluding English:¹⁵

Rule 6:

```
ls: ?Xss {dpos=N
      det->?Yss}
rs: ?Ytp{gender=?Xss.gender
      number=?Xss.number}
cr: ?Xss⇔?Xtp; ?Yss⇔?Ytp
cd: not(lexicon:: (id).(iso)=ENG)
```

Another option would be to provide this rule family-wise, specifying in the conditions `language:: (id).(family)=romance`.

The general principle is thus to minimize the number of language-specific rules (such as Rule 3) and maximize the number of generic rules. The degree of generalization that can be achieved for a module depends on the language and the strata that are involved. Languages with a complex agreement or many lexical markers for

¹⁵ For instance, Dutch forms in this respect a common set with English. This means that if we would include Dutch into our resource list, we would need to modify the rule accordingly.

grammatical meanings (articles, auxiliaries, etc.) require a higher share of language-specific rules, while languages with a poorer morphology have a smaller share of language-specific rules.

Table 3 below copied from [10] shows the distribution of the language-independent and language-specific rules in a medium coverage generation grammar system that involves Catalan (CT), English (EN), Spanish (ES), French (FR), Polish (PL) and Portuguese (PT).

Table 3: Distribution of generic and language-specific rules in a generation grammar framework

Module	Core	CT	EN	ES	FR	PL	PT
Con $\Rightarrow$ Sem	50	0	0	0	0	0	0
Sem $\Rightarrow$ DSynt	59	8 (12%)	8 (12%)	7 (11%)	7 (11%)	11 (16%)	6 (9%)
DSynt $\Rightarrow$ SSynt	64	13 (17%)	16 (20%)	11 (15%)	16 (20%)	7 (10%)	12 (16%)
SSynt $\Rightarrow$ DMorph	70	13 (16%)	3 (4%)	12 (15%)	19 (21%)	8 (10%)	14 (17%)
DMorph $\Rightarrow$ SMorph	7	8 (53%)	5 (42%)	6 (46%)	10 (59%)	10 (59%)	6 (46%)
SMorph $\Rightarrow$ Text	12	1 (8%)	1 (8%)	1 (8%)	1 (8%)	1 (8%)	1 (8%)
Language-specific rules (%), avr.		14%	(11%)	(12%)	(17%)	(12%)	(13%)

As can be observed, the deeper the stratum, the more generic a grammar module tends to be. Thus, the Con-Sem module is entirely language-independent, but highly domain-specific – certainly also because the above six languages are very similar with respect to their semantic codification.

5.2. Packaging language-specific rules

For further optimization of the organization of the resources, each grammar module can be organized in terms of packages. A package is a set of rules that complement each other, or, on the contrary, are in competition. Each package handles one specific linguistic phenomenon. Formally, a package is defined by an abstract rule from which other rules depend. An abstract rule is always empty, but it may have conditions associated to it. These conditions are then inherited to all rules that depend on it. A package can be also composed of several sub-packages – as, for instance, the language packages: rules specific to a language are grouped into a separate package, which in its turn consists of a number of sub-packages for various language-specific phenomena.

In each grammar module, a “core” package is defined. It contains all essential rules that are needed to process a standard input structure. For instance, the core package of the SSynt-DMorph module includes rules for lexicalization and actantial and modifier relation handling. Packages for lexical co-occurrence, quantification, voice, etc. complement the core package.

With this package-oriented design, it is possible to: (i) localize easier fragments of grammatical resources that can be shared across languages, (ii) exchange fragments of the resources without worrying about the interdependencies with other fragments, (iii) assign the development of different packages to different grammarians, observing the expertise and interests of each of them.

The strategy of packaging can also be extended to the design of dictionaries. Thus, the lexical core should constitute the main package, while additional packages could be developed by terminologists as experts in specific areas of specialized discourse. Furthermore, sub-packages could be introduced for each zone of a dictionary (see Section 2.3 above) and for semantic/lexical fields or domains.

5.3. *Shifting the bulk of the grammarian's work to the lexicon*

One of the obstacles during the development of large coverage generation grammars – be it for low-density or high-density languages – is the complexity of the transductions if they are considered as a purely grammatical matter. One way to ease the work of grammar writing is to encode information that is in its nature lexical in the dictionary for the individual LUs. For instance, the mapping of the argument relations in a semantic and deep-syntactic structure onto corresponding relations or node configurations at the adjacent stratum has in former MTT-generators been usually realized in terms of grammar rules of the following type:

Rule 6*:

```
ls: ?Xsem {1->?Ysem}
rs: ?Xds{I->?Yds}
cd: ?Ysem.theme = yes
cr: Xsem⇔?Xds; ?Ysem⇔?Yds
```

Rule 7*:

```
ls: ?Xsem {2->?Ysem}
rs: ?Xds{II->?Yds}
cd: ?Ysem.rheme = yes
cr: Xsem⇔?Xds; ?Ysem⇔?Yds
```

However, these rules ignore that the correspondences they realize are the default projections as encoded in the government pattern (GP) of any predicate and should thus be specified in the dictionary in a general form:

```
"predicate" {
  gp={1=I; 2=II; 3=III; ...; 6=VI}}
```

We assume that a predicate possesses at most six arguments, with the *i*-th semantic argument (denoted by an Arabic numeral) by default corresponding to the *i*-th syntactic actant (denoted by a Roman numeral).

Each individual predicative LU inherits this GP such that a repetition is avoided (see also next subsection). In case an LU has a deviating projection, this deviation is entered in its individual GP. Grammar rules can then make use of the GP information (both inherited and individual) as indicated in Rule 2 in 4.1.1.1.

Such a shift of the workload from the grammar to the lexicon is also in accordance with our experience, which tells us that for untrained personnel it is much easier and faster to write lexical entries than to write grammar rules.

5.4. Relying upon rich hierarchical dictionaries

As already pointed out above, rich dictionaries are crucial for the development of stable large coverage MTT-based generators. The information that needs to be included in such dictionaries has already been outlined in Section 4.1.2. For an efficient organization of this information, we use inheritance. The inheritance allows us to factorize all information shared by several entries into abstract entries, from which it is then propagated to the entries for the concrete LUs. Consider a fragment of the verbal hierarchy “predicate” → “verb” → “direct transitive verb”. The predicate node provides the default projection of the semantic valency to the syntactic valency of a predicative unit; cf. above. A verbal lexeme is a predicate (i.e., inherits, if they are not overwritten, all features defined for the predicate unit). Furthermore, its surface and deep part of speech are respectively ‘V’ and ‘verb’ and, by default, in the case of the finite form, its first syntactic actant is realized as a syntactic subject, usually a noun:

```
"verb" : "predicate"{
dpos=V; spos=verb
gp={I={dpos=N; rel=subj}}}
```

Note that such abstract entries are not necessarily universal. For each language, we keep a separate hierarchy since the parts of speech and the morpho-syntactic behavior of the lexical units that carry them can vary cross-linguistically. For example, in Polish, the syntactic subject is assigned the nominative case, such that this information would be added to the abstract verb entry for Polish; the exceptions are taken care of within the entries for the verbs in question.

English direct transitive verbs inherit from the “verb”-class. Furthermore, they realize their second syntactic actant as a direct object, and it is by default a noun:

```
"verb_dt" : "verb" {
gp={II={dpos=N; rel=dobj}}}
```

Now, adding a direct transitive verb to the lexicon is just a matter of expressing its membership in the verb_dt class and adding verb-specific information. Consider, for illustration, the entry for the verb EXCEED, where we have added information on its lexical co-occurrence:

```
exceed : "verb_dt"{
Magn="by far"
Magn="far"
AntiMagn="a little"}
```

All information on the projection of the semantic to the syntactic valency, the part of speech and the surface realization of the actants has been inherited. This information can be overwritten. For example, [to] EXPECT has two possible sub-categorization patterns, none of which corresponds to the default pattern for verbs:

```
expect : "verb" {
gp={ II={dpos=V; finiteness=FIN; mood=IND; prep=that}}
gp={II={dpos=V; finiteness=INF; prep=to; rel=iobj}
raise={II={rel=dobj}}}
```

The first pattern gives rise to such sentences as *[We]_I expect [that the ozone concentration will increase]_{II}*. The second pattern encodes a subject-raising construction where the first actant of DSynt-actant **II** is raised to become the direct object of EXPECT, downgrading actant **II** to an indirect object position; cf.: *[We]_I expect [the ozone concentration]_{II/raised subject, dir.obj} [to increase]_{III}*.

6. Deriving Generation Resources for New Languages

The resource design principles discussed in Section 5 ensure that (i) no parts of the resources are repeated, (ii) the resources are linguistically sound, and (iii) the acquisition and maintenance (evaluation, correction, and extension) of the resources can be carried out easily by grammarians without extensive experience in the linguistic theory underlying the generator. Especially the last item is of relevance to our context. The principles allow for a swift acquisition of resources for new languages – including low- and middle density languages. Few changes need to be made to the grammar rules, since most of the language-specific information is dealt with in the dictionaries. Rules that handle articles, auxiliaries and other lexical markers of grammatical meanings, as well as agreement and word-order rules do need to be modified, but they are usually rather simple. Hence, the major workload of adding a new language consists in the description of the LUs of the language in question – although the distribution of the workload obviously heavily depends on the characteristics of the new language. Thus, the extension of the resources developed for the six languages Catalan, English, French, Portuguese, and Spanish to Galician,¹⁶ which is very similar to Spanish, hardly required any new grammar rules; the main workload consisted in the translation of the vocabulary, i.e., in porting of the lexical resources. Consider the Galician lexical entries corresponding to EXCEED and EXPECT from above:

```
exceder: "verb_dt"{
Magn=moito
AntiMagn="un pouco"}

esperar : "verb" {
gp={II={dpos=V; finiteness=FIN; mood=SUBJ; prep=que}}
gp={II={dpos=V; finiteness=INF; prep=a; rel=iobj}
raise={II={rel=dobj}}}}
```

Note the difference in the mood of ESPERAR ‘expect’: while in English, it is the indicative, in Galician (as in Catalan and Spanish) it is the subjunctive.

The extension of the same resources to, e.g., Basque, would be more costly, and to, e.g., Lezgian (just to name a language that is rather different from the languages with which we usually work), or any other typologically radically different language, even more so.¹⁷

¹⁶ Galician is a Western Ibero-Romance language spoken in the North-West of Spain; the total number of speakers of Galician is about 3 to 4 million.

¹⁷ According to Wikipedia, Basque has about 1 million speakers (for 700 000 of them, it is the first language) and Lezgian, which belongs to the Lezgic group of the Dagestan language family, about 451 000 speakers.

7. Evaluation of the Resources

Any language engineering resources must be subjected to a rigorous evaluation at each stage of their development. This is also true for resources being obtained by resource sharing and porting. Lareau and Wanner [10] propose for MTT-based generation resources a three stage evaluation: (i) micro evaluation of grammatical resources, (ii) macro evaluation of grammatical resources, and (iii) evaluation of the consistency of the dictionaries.

Micro evaluation concerns the evaluation of individual rules. For each rule (or each set of equivalent rules) a set of test structures is to be set up. These structures must be as simple as possible but still cover all phenomena the rule in question has to be able to cope with. In most cases, a single rule cannot be tested in isolation; its application depends on the application of some core rules. For instance, it is not possible to test only the construction of a given syntactic relation without also applying the rules that create the nodes linked by the relation. Therefore, for evaluation, it is necessary to keep track of rule dependencies. Hence, not only do we associate a set of test structures with each rule, but we also associate each test structure with the set of rules it activates. When a rule is modified, all executed test routines that involved it are reset and run again. Micro-testing thus verifies elementary components separately.

By the very nature of micro evaluation, it is difficult to have for it language-independent test structures. Thus, even if we want to test a generic DSynt-SSynt rule, the input test structure will have to be a DSynt-structure, which by definition contains LUs of a specific language. Therefore, only the rules of the Con-Sem module can be micro-tested with language-independent test structures (since our conceptual structures are the same for all languages) – although it can often be assumed that generic rules tested in one language will work just as well in other languages.

Macro evaluation aims to assess the coverage of the linguistic resources for a given application – as, for instance, the generation of air quality bulletins or soccer match commentaries. The test structures used for this task must cover the largest possible number of text plans the generator has to handle. The goal is here not to test specific rules, but to make sure that the system can handle the input structures with which it is expected to be provided. Macro evaluation is best applied after micro evaluation, as it verifies the interaction between the various components of the grammar.

In addition to micro and macro evaluation of the grammatical resources, it is necessary to make sure that all units of the node alphabet at a given stratum can be expressed in terms of a feature, unit, or unit configuration at the adjacent stratum. In the context of text generation which covers all strata of an MTM, we must also ensure that all concepts that might appear in the input structures can be expressed in any language. Concepts are mapped following the instructions in the conceptual dictionaries to language-specific semantemes, which in their own turn are mapped to LUs. These LUs point to prepositions in their sub-categorization patterns and to other LUs through lexical functions. All these links form a complex network where errors are hard to spot for a human, so we created a small MATE grammar that consists essentially of simplified lexicalisation rules. This grammar takes as input a list of structures, each containing one concept (one structure for every concept expected in the input of the system), and produces structures representing the lexical links encoded by the dictionaries. Then, a set of consistency-checking rules is applied to make sure that there is no pointer to non-existent entries, and that each entry contains all the necessary

information (for example, that Catalan nouns have a gender, that every LU has a part of speech, that syntactic relation names are specified in the sub-categorization patterns, etc.). If an error is found, an appropriate message is added to the output.

8. Conclusions

We hope that it became clear why MTT is one of the most popular linguistic theories in text generation. Our goal in this article was twofold: firstly, to give a brief introduction to the MTT-model and to its use as basis of a text generator, and, secondly, to show how grammatical and lexical resources in the MTT-framework can be efficiently developed and extended to cover new languages – be they high-, or low-, or middle-density languages. However, given that so far we experimented only with low- and middle-density languages that are very similar to the high-density languages for which resources are available, we cannot make any reliable statement on how big the effort would be to extend our resources to radically different languages. There is no other way to know it than to get started. Researchers embarking on this endeavour can be sure of our support.

Acknowledgements

This paper has been read and commented upon by M. Alonso Ramos and I. Mel'čuk; many thanks to both of them for valuable comments and suggestions, which helped to improve the first version of the paper considerably. The remaining errors are, as always, our own responsibility.

References

- [1] Mel'čuk, I. *Opyt teorii lingvističeskix modelej "Smysl-Tekst"*. Nauka, Moscow, 1974.
- [2] Mel'čuk, I. *Dependency Syntax*. State University of New York Press, Albany, 1988.
- [3] Mel'čuk, I. *Vers une linguistique Sens-Texte. Leçon inaugurale*. Collège de France, Paris, 1997.
- [4] Kahane, S. The Meaning-Text Theory. In: V. Agel et al. (eds.). *Dependency and Valency. An International Handbook of Contemporary Research*, Vol. 1, 546-570. De Gruyter, Berlin/New York, 2003.
- [5] Mel'čuk, I. *Communicative Organization in Natural Language*. Benjamins Academic Publishers, Amsterdam, 2001.
- [6] Mel'čuk, I. and L. Wanner. Syntactic mismatches in machine translation. *Machine Translation* **20** (2006), 81-138.
- [7] Mel'čuk, I. and L. Wanner. Morphological mismatches in machine translation. Submitted.
- [8] Bohnet, B. Covering the Mapping from Semantics up to Morphology by Transducers. In K. Gerdes et al. (eds.) *Meaning-Text Theory 2007*, 129-138. Sagner, Munich, 2007.
- [9] Bohnet, B. and L. Wanner. On Using a Parallel Graph Rewriting Formalism for Text Generation. In *Proceedings of the 8th European Natural Language Generation Workshop at the Annual Meeting of the Association for Computational Linguistics*. Toulouse, 47-56. 2001.
- [10] Lareau, F. and L. Wanner. Towards a Generic Multilingual Dependency Grammar for Text Generation. In T. Holloway King and E.M. Bender (eds.) *Proceedings of the GEAF 2007 Workshop. CSLI Studies in Computational Linguistics* ONLINE. <http://csli-publications.stanford.edu>. Stanford, 2007.
- [11] Halliday, M.A.K. *An Introduction to Systemic Functional Grammar*. Arnold, London, 1994.
- [12] Halliday, M.A.K. and C. Matthiessen. *Construing Experience through Meaning: A Language Based Approach to Cognition*. Cassell, London, 1999.
- [13] Dik, S.C. *The Theory of Functional Grammar*. Foris, Dordrecht, 1989.

- [14] Croft, W. and D.A. Cruse. *Cognitive Linguistics*. Cambridge University Press, Cambridge, 2004.
- [15] Langacker, R.W. *Foundations of Cognitive Grammar Vols. I, II*. Stanford University Press, Stanford, 1987/1991.
- [16] Van Valin, R.D. *Advances in Role and Reference Grammar*. Benjamins Academic Publishers, Amsterdam, 1993.
- [17] Matthiessen, C.M.I.M. and J.A. Bateman. *Text Generation and Systemic Functional Linguistics*. Pinter Publishers, London, 1991.
- [18] Miličević, J. *La paraphrase*. Peter Lang, Berne, 2007.
- [19] Mel'čuk, I. Phrasemes in Language and Phraseology in Linguistics. In M. Everaert et al. (eds.) *Idioms*, 167-232. Lawrence Erlbaum Associates, Hillsdale, NJ, 1995.
- [20] Mel'čuk, I. Lexical Functions: A Tool for the Description of Lexical Relations in a Lexicon. In L. Wanner (ed.) *Lexical Functions in Lexicography and Natural Language Processing*, 37-102. Benjamins Academic Publishers, Amsterdam, 1996.
- [21] Mel'čuk, I., A. Polguère and A. Clas. *Introduction à la lexicologie explicative et combinatoire*. Duculot, Louvain-la-Neuve, 1995.
- [22] Alonso Ramos, M. *Las funciones léxicas en el modelo lexicográfico de I. Mel'čuk*. PhD Thesis. U.N.E.D., Madrid, 1993.
- [23] Mel'čuk, I. and L. Wanner. Towards a lexicographic approach to lexical transfer in machine translation (illustrated by the German-Russian language pair). *Machine Translation* **16** (2001), 21-87.
- [24] Gerdes, K. and S. Kahane. Phrasing it Differently. In L. Wanner (ed.) *Selected Lexical and Grammatical Issues in the Meaning-Text Theory*, 297-335. Benjamins Academic Publishers, Amsterdam, 2007.
- [25] Engel, U. *Deutsche Grammatik*. Julius Gross, Heidelberg, 1988.
- [26] Daneš, F. Functional Sentence Perspective and the Organization of the Text. In F. Daneš (ed.) *Papers on Functional Sentence Perspective*, 106-128. Academia, Prague, 1974.
- [27] Sgall, P., E. Hajičová, and J. Panevová. *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*. Reidel Publishing Company, Dordrecht, 1986.
- [28] Lambrecht, K. *Information Structure and Sentence Form. Topic, Focus, and the Mental Representation of Discourse Referents*. Cambridge University Press, Cambridge, 1994.
- [29] Erteschik-Shir, N. *Information Structure. The Syntax-Discourse Interface*. Oxford University Press, Oxford, 2007.
- [30] Wanner, L. and M. Alonso Ramos. What Type of Entity Is a Lexical Function? In *Proceedings of the 2nd International Conference on the Meaning-Text Theory*, 518-528. Moscow, 2005.
- [31] Lavoie, B. and O. Rambow. A fast and portable realizer for text generation systems. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, 265-268, 1997.
- [32] Coch, J., E. De Dycker, J.-A. Garcia-Moya, H. Gmoser, J.-F. Stranart and J. Tardieu MultiMeteo: adaptable software for interactive production of multilingual weather forecasts. In: *Proceedings of the 4th European Conference on Applications of Meteorology (ECAM 99)*. Norköping, Sweden, 1999.
- [33] Wanner, L., B. Bohnet, N. Bouayad-Agha, F. Lareau, A. Lohmeyer, and D. Nicklaß. On the Challenge of Creating and Communicating Air Quality Information A Case for Environmental Engineers. In *Proceedings of Environmental Software Systems: Dimensions of Environmental Informatics*. Prague, 2007.
- [34] Nicklaß, D., N. Bouayad-Agha and L. Wanner. Addressee-Tailored Interpretation of Air Quality Data. In *Proceedings of EnviroInfo*, Vol. 2, 67-74, 2007.
- [35] Sowa, J. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks Cole Publishing Co, Pacific Grove, CA, 2000.
- [36] McKeown, K. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press, Cambridge, 1985.
- [37] Mann, W.C. and S. Thompson. Rhetorical Structure Theory: A theory of text organization. In L. Polanyi (ed.) *The Structure of Discourse*. Ablex Publishing Corporation, Norwood, New Jersey, 1987.
- [38] Schank, R. & R. Abelson. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1977.
- [39] Hovy, E. Automated Discourse Generation Using Discourse Structure Relations. *Artificial Intelligence* **63** (1993), 341-386.
- [40] Moore, J. and C. Paris. Planning Text for Advisory Dialogues: Capturing Intentional and Rhetorical Information. *Computational Linguistics* **19** (1993), 651-694.
- [41] Rösner, D. and M. Stede. Customizing RST for the Automatic Production of Technical Manuals. In R. Dale et al. (eds.) *Aspects of Automated Natural Language Generation*. Springer Verlag, Berlin, 1992.
- [42] Bouayad-Agha, N. and L. Wanner. Text Planning of Air Quality Information. In *Proceedings of EnviroInfo 2007*, Vol. 2, 81-88, 2007.

- [43] Bouayad-Agha, N., D. Nicklaß and L. Wanner. Discourse Structuring of Dynamic Content. In *Proceedings of the Spanish Conference on Computational Linguistics*. Zaragoza, 2006.
- [44] Dalianis, H. Aggregation in Natural Language Text Generation. *Computational Intelligence*, **15** (1999), 384-404.
- [45] Shaw, J. *Clause Aggregation: An Approach to Generating Concise Text*. PhD Thesis. Columbia University, New York, 2002.
- [46] McKeown, K. and M. Derr. Using Focus to Generate Complex and Simple Sentences. In *Proceedings of the 10th International Conference on Computational Linguistics*, 501-504, 1984.
- [47] Iordanskaja, L., R. Kittredge, and A. Polguère. Lexical Selection and Paraphrase in a Meaning-Text Generation Model. In C.L. Paris et al. (eds.). *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, 293-312. Kluwer Academic Publishers, Dordrecht, 1991.
- [48] Wanner, L. *Exploring Lexical Resources for Text Generation in a Systemic Functional Language Model*. PhD Thesis. Saarland University, Saarbrücken, 1997.
- [49] Wanner, L. Lexicalization in Text Generation and Machine Translation. *Machine Translation*, **11** (1996), 3-35.
- [50] Reiter, E. A New Model for Lexical Choice for Nouns. *Computational Intelligence*, **7** (1991), 240-251.
- [51] McDonald, D. On the Place of Words in the Generation Process. In C. Paris et al. (eds.) *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, 227-247. Kluwer Academic Publishers, Dordrecht, 1991.
- [52] Stede, M. *Lexical Semantics and Knowledge Representation in Multilingual Text Generation*. Kluwer Academic Publishers, Dordrecht, 1999.
- [53] Polguère, A. A "Natural" Lexicalization Model for Language Generation. *Proceedings of the Fourth Symposium on Natural Language Processing 2000*, 37-50. Chiangmai, Thailand, 2000.
- [54] Bohnet, B. and R. Dale. Viewing Referring Expression Generation as Search. In the *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*. Edinburgh, Scotland, 2005.
- [55] Horacek, H. On Referring to Sets of Objects Naturally. In H. Bunt and R. Muskens (eds.). *Third International Natural Language Generation Conference*, 70 – 79. Springer Verlag, Heidelberg, 2004.
- [56] Krahmer, E., S. van Erk, and A. Verleg. Graph-based generation of referring expressions. *Computational Linguistics*, **29** (2003), 53–72.
- [57] Dale, R. *Generating Referring Expressions: Constructing Expressions in a Domain of Objects and Processes*. MIT Press, Cambridge, MA, 1992.
- [58] Aho, A.V. and J.D. Ullman. Translations of a Context-Free Grammar. *Information and Control*, 1971.
- [59] Bohnet, B., A. Langjahr and L. Wanner. A Development Environment for an MTT-Based Sentence Generator. *Proceedings of the First International Conference on Natural Language Generation*. Mitzpe Ramon, Israel, 260-263, 2000.
- [60] Bohnet, B. *Textgenerierung durch Transduktion linguistischer Strukturen*. AKA, Berlin, 2006.
- [61] Goldberg, E., N. Driedger, and R. Kittredge. Using Natural Language Processing to Produce Weather Forecasts. *IEEE Expert*, **9** (1994), 45-53.
- [62] Bethem, T., J. Burton, T. Caldwell, R. Kittredge, B. Lavoie, and J. Werner. Generation of Real-Time Narrative Summaries of Real-time Water Levels and Meteorological Observations in PORTS®. In *Proceedings of the Fourth Conference on Artificial Intelligence Applications to Environmental Sciences*, San Diego, California, 2005.
- [63] Coch, J. and R. David. Representing Knowledge for Planning MultiSentential Text. In *Proceedings of the 4th Conference on Applied Natural Language Processing*, 1994.
- [64] Wanner, L., B. Bohnet, and M. Giereth. Deriving the Communicative Structure in Applied NLG. *Proceedings of the 9th European Natural Language Generation Workshop at the Annual Meeting of the Association for Computational Linguistics*, Budapest, 111-118, 2003.
- [65] Alshawi, H. *The Core Language Engine*. The MIT Press, Cambridge, MA, 1992.
- [66] Kim, R. M. Dalrymple, R.M. Kaplan, T. Holloway King, H. Masuichi, and T. Ohkuma. Multilingual Grammar Development via Grammar Porting. In *Proceedings of ESSLLI Workshop on Ideas and Strategies for Multilingual Grammar Development*, 2003.
- [67] Avgustinova, T. and H. Uszkoreit. An ontology of systemic relations for a shared grammar of Slavic. In *Proceedings of the 18th International Conference on Computational Linguistics*, 28-34, 2000.
- [68] Bender, E., D. Flickinger, and S. Oepen. The Grammar Matrix: An Open-Source Starter-Kit for the Rapid Development of Cross-Linguistically Consistent Broad-Coverage Precision Grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, 8-14, 2002.
- [69] Bateman, J., I. Kruiff-Korabayova, and G.J. Kruiff. Multilingual resource sharing across both related and unrelated languages. An implemented, open-source framework for practical natural language generation. *Journal on Language and Computation*, **3** (2005), 191-219.